04

Validando duplicados

Transcripción

[00:00] Vamos a crear aquí, un mecanismo en nuestro código para evitar que los números aleatorios generen números repetidos. Lo vamos a hacer aquí dentro de nuestro while, y vamos a aplicar una iteración dentro de otra iteración. En este caso, vamos a usar un for para mostrar que los dos se pueden usar sin ningún problema y vamos a pensar un poco qué es lo que necesitamos hacer.

[00:37] Bueno, tenemos nuestra iteración del while, nuestro contador, en este caso, que va a iterar cuantas veces me diga la cantidad. Si son 4, va a iterar 4, si son 3, va a iterar tres veces. Ahora, en cada una de esas iteraciones yo necesito entrar ya al arreglos secretos y verificar los valores que ya fueron cargados en las etapas anteriores.

[01:05] Si estoy en la segunda rodada, tengo que verificar lo que fue cargado en la primer rodada y ver si un número aleatorio que generé en la segunda no fue cargado en la primera. Para eso, vamos a hacer una iteración. Aquí vamos a definir una variable que lo vamos a llamar de posición, lo vamos a iniciar con 0, y en nuestro for, la segunda parte es la condición donde le vamos a decir que posición sea menor que secretos.length. Eso ya lo habíamos visto que es el tamaño de secretos.

[01:47] Y aquí nuestro contador. Entonces, en cada iteración, primero secreto va a tener una posición, después va a tener dos, después va a tener tres, y así sucesivamente. Y aquí nuestro for. Entonces, ¿qué vamos a poner aquí en

nuestro for? Vamos a hacer una condición donde le vamos a decir que si número aleatorio es igual a secretos posición, que nos ejecute lo siguiente.

[02:34] Para ello, vamos a crear una variable, aquí nos olvidamos de escribir variable, aquí vamos a crear una variable que lo vamos a llamar encontrado, va a ser un booleano y lo vamos a inicializar con false. Entonces, si encontramos un número aleatorio igual a los elementos que ya fueron cargados, le vamos a decir que encontrado sea true.

[03:18] Entonces, si número aleatorio es igual a secretos, con la posición le vamos a decir que sea true. Entonces cuando acaba el for, una vez que acaba este for de pasar por todos los valores que ya tienen secretos, aquí le vamos a decir si eso encontrado es igual a false, que me ingrese los valores a mi variable.

[04:03] Ahorita vamos a repasar lo que hemos hecho. ¿Qué hicimos? Creamos una iteración con for en la cual estamos diciendo que iteremos cuantas veces ya hayamos cargado en nuestro arreglo secretos. Entonces, si está en la primera iteración, aquí va a decir posición es 0, secretos length no tiene ningún tamaño, o sea todavía no cargamos nada ni va a entrar al for y va a volver de nuevo.

[04:36] En este caso, ya secretos length va a tener una posición, 0 menor que 1, va a entrar aquí, va a comparar si lo que cargamos en la primer rodada es igual al nuevo número aleatorio generado, si si, entra aquí, nos niega esto, no va a entrar, no va a añadir ese número aleatorio y va a generar otro número aleatorio.

[05:01] Entonces, yo puedo generar 10 números aleatorios, pero voy a ingresar solo la cantidad de números sin repetir que yo estoy mandando aquí en mi parámetro que en este caso va a ser 4. Aquí vale destacar que este contador no puede estar aquí. ¿Por qué? Vamos a pensar. Si yo mando aquí y yo encuentro

que tuve repetidos, no puedo contabilizar mis elementos, no puedo decirle que ingresé un elemento.

[05:40] ¿Concuerdan conmigo? Porque ese contador tiene que estar dentro de este if, si y solo si no encontré ningún repetido, yo puedo añadir un valor y contabilizar para rodar mi while. Si este contador estuviera aquí afuera, el efecto va a ser que voy a rodar, voy a encontrar repetidos y voy a ingresar menos valores, porque si no entro aquí, no ingreso ningún número y contabilizo +1, las iteraciones van a ser menores que la cantidad de elementos que yo voy a ingresar.

[06:22] Entonces, por eso este contador tiene que estar dentro de mi if aquí. Para entender un poco lo que está haciendo nuestro programa, aquí voy a colocar también un log para que veamos qué números aleatorios estamos generando y así vamos a entender lo que nuestro programa está haciendo.

[06:45] Entonces, vamos a repasar rápidamente. Voy a llamar sorteos, lo voy a pasar como parámetro número 4, y aquí va a comenzar este loop while, el contador empieza en 1 hasta 4. Y aquí va a comenzar a generar primero un número aleatorio me va a llamar a la función aleatoria.

[07:07] Inicializamos esto como falso, encontrado igual a falso. Inicializamos un loop interno dentro de nuestro while para validar que en nuestros arreglos secretos no haya ningún valor aleatorio, no haya el nuevo número aleatorio que estamos generando en cada una de las rodadas del while y hayamos cargado dentro del arreglo.

[07:30] Para eso está este trigger que estamos colocando aquí o esta bandera, que en este caso, es booleano, puede ser falso o verdadero. Aquí una recomendación que ya la habíamos visto, caso ingrese, no salimos de este for de aquí. Va a mirar este break, recordando que es como si fuera una tijera que corta el último loop, que va a ser este for de aquí.

[08:04] Vamos a guardar y voy a ir aquí. Aquí es solo un minuto. Entonces, voy a actualizar. ¿Qué nos generó? Nos generó el 9, el 1, el 4 y el 5. Ningún valor repetido. Ahora sí, fíjense, nos generó un 10, un 7, un 4, el 7 lo repitió dos veces pero solo me ingresó uno aquí. El 4 me generó tres veces, pero solo me ingresó uno aquí.

[08:32] Entonces, nuestra función de eliminar repetidos está funcionando a la perfección. 6, 5, 0, 5 y 1. Nos repitió el 5, ya no lo ingresó, solo que aquí nos está mostrando el 0 y era una de las especificaciones de nuestro programa que queremos números del 1 al 10, no del 0 al 10.